REMARKS

The Examiner rejected claims 1, 3-6, 11-15, 17-19, 25, and 27-29 pursuant to 35 U.S.C. §103(a) as unpatentable over Smith, et al. (US 6,241,675) in view of Park (US 6,192,164). Claim 2 was rejected pursuant to 35 U.S.C. §103(a) as unpatentable over Smith, et al. in view of Park, and further in view of Zar (A Scan Conversion Engine...). Claims 7 and 20 were rejected pursuant to 35 U.S.C. §103(a) as unpatentable over Smith, et al. in view of Park and Drebin, et al. (US 4,835,712). Claims 9 and 22 were rejected pursuant to 35 U.S.C. §103(a) as unpatentable over Smith, et al. in view of Park and Swerdloff (US 5,483,567). Claims 24 and 26 were rejected pursuant to 35 U.S.C. §103(a) as unpatentable over Smith, et al. in view of Park and Halmann (US 6,526,163).

Claims 10 and 23 were allowed. Claims 8 and 21 were objected to as being allowable if amended into independent form.

Applicants respectfully request reconsideration of the rejections of claims 1-7, 11-20, 22, and 24-29, including independent claims 1, 14, 28, and 29. *New remarks are added below in italics.*

Independent claim 1 recites a processor operable to interpolate where the interpolation is a function of three-dimensional rendering rays cast through the virtual volume, wherein the processor is configured to avoid scan-conversion from the polar coordinates of volume data that does not contribute to a final volume rendered image by scan converting only visible voxels based on the rendering rays and corresponding Cartesian coordinates, the identifying corresponding to identifying Cartesian coordinates associated with visible voxels of the final volume rendered image. Smith, et al. and Park do not disclose these limitations.

Park is cited for use of a look-up-table in scan conversion (Office Action, pages 4 and 5). Two-dimensional scan conversion is provided (col. 3, line 58; and see Figs. 1 and 2). Unit distance is used to scan convert for proceeding through the data (col. 4, lines 30-36). Park does not interpolate as a function of 3D rendering rays through a virtual volume. Park does not avoid scan-conversion for some data. Park does not identify associated with visible voxels in a volume rendered image.

Smith, et al. form 3D data sets (col. 1, lines 59-64; and col. 3, lines 55-65). The 3D data is used for generating slice images (col. 2, lines 10-18). The user selects slices in the volume (col. 11, lines 57-60; see Figs 1a-i). A scan converter produces an image selected by the user (col. 19, lines 40-44). X, y, z points are interpolated by stepping along the x, y and z dimensions in unit steps (col. 20, lines 62-66). These steps are along linear rays for horizontally producing raster lines for a slice plane (col. 20, line 62-col. 21, line 3). The raster scan moves down lines of a frame for generating any arbitrary slice in 3D space (col. 21, lines 3-5). Smith, et al. scan converts one or more planes from a scanned volume using the plane position in 3D. Smith, et al. do not scan convert as a function of three-dimensional rendering rays cast through a virtual volume. The linear rays of Smith, et al. are display raster lines, not rays cast as part of volume rendering.

Smith, et al. use data based on user positions of slices, so may avoid scan conversion of data not along the slices. However, Smith, et al. rely on slice position, not whether the voxel is visible based on rendering rays and not by identifying Cartesian coordinates associated with visible voxels of the final volume rendered image. Smith, et al. determine the locations of the slices based on user positioning in the volume.

In response, the Examiner notes that Smith, et al. operate from a starting point and trace down 3d linear rays by applying dxdu, dydu, dzdu as a horizontal raster line is produced, alleging that this is discussing ray tracing in 3D, not simply rays that are one-dimensional raster lines and alleging that the horizontal raster line refers to the display created by ray tracing. However, this reading ignores the context and specific statements of Smith, et al. Throughout the specification, Smith, et al. discuss the identification and viewing of planes or slices in the volume (see col. 5, lines 3-36, 51, 56, 59-64; col. 6, lines 8, 17, 31, 34, 43, 46, 46, 51, 52, 56, 57, 59-66; col. 11, lines 25 and 59; col. 12, line 28; col. 20, line 67; col. 21, line 5; and col. 26, lines 29, 32-36). Figures 1a-1i show such slices. This overwhelming mentioning of imaging slices or planes within the scanned volume indicates that Smith, et al. are teaching scan conversion for a slice, not three-dimensional rendering rays cast through the volume, at least in the scan conversion discussion of columns 19-21.

Smith, et al. is concerned with finding an aiming line for a point (col. 1, lines 38-47;

and col. 5, lines 54-64). To find the aiming line, two-dimensional images from slices in the volume are used (col.5, lines 54-64). The user can reposition the slices in the volume to find the desired line as an intersection of slices (col. 5, lines 59-64). To fulfill the purpose of Smith, et al., two-dimensional images of slices in the volume are output. This indicates that the scan conversion discussion (columns 19-21) is for generating a two-dimensional slice image, not three-dimensional rendering rays cast through the volume.

The use of a user interface to select the slices in the volume shows that these 3D linear rays are not three-dimensional rendering rays. The scan converter is noted as being used to produce views selected by the user via the interface 210 (col. 19, lines 41-48). This interface 210 is described as being for user selection of B slices (col. 11, lines 57-65). Since the disclosure in column 20 cited by the Examiner describes operation of this scan converter for user selected slices, this context indicates operation to interpolate data for a slice, not a three-dimensional rendering.

Typically, a person of skill in the art uses scan conversion to refer to conversion of polar coordinates to Cartesian coordinates in imaging a plane or two-dimensional imaging. For three-dimensional imaging, a three-dimensional processor or other terminology is used. The disclosure at col. 20 is directed to scan conversion, indicating slice imaging and not three-dimensional rendering rays.

In addition to the overwhelming context indicating any rays being for two-dimensional slice viewing, the specific disclosure for the scan converter in columns 19-21 indicates usage for two-dimensional slice imaging. The use of change in x, y, and z is merely the result of any arbitrary plane in a volume intersecting different locations in all three dimensions (see col. 26, lines 25-30). Since cubic interpolation is used for scan conversion, the slices adjacent a given point are converted for interpolating the point on the plane to be viewed (col. 26, lines 25-42). Figures 17 and 18 show this process described in column 26 for a point in the slice being imaged (Fig. 18) (col. 30, lines 58-59). The use of x, y, and z is for imaging a slice, as mentioned specifically at col. 20, line 67 and col. 21, line 5.

The mention of "3D linear rays" does not change that the process is for identifying points in a slice or plane for two-dimensional imaging of the plane. The 3D linear rays at *column 20, line 65 are used as the display raster line is horizontally populated (col. 20, lines 62-66). As discussed above, the display is for a slice image. This is reinforced by the statement at column 21, lines 3-6 that the raster scan traverses in x, y, and z to "generate any arbitrary slice in 3D space." The 3D linear rays are not three-dimensional rendering rays. The 3D linear rays merely define the plane location. As such, they are not three-dimensional rendering rays. To the extent that the rays are in the data volume, it is to find the plane, not for three-dimensional rendering. The discussing at column 20, line 56-column 21, line 11 is for imaging an arbitrary slice as desired by Smith, et al. given their purpose, not for three-dimensional rendering.*

Even if Smith, et al. provide for volume rendering, there is not disclosure of the type of volume rendering used. Various types of volume rendering may be provided without using ray casting.

Since Smith, et al. scan convert for a plane or slice, the 3D linear rays are for defining a two-dimensional plane in a volume. The 3D linear rays are not rays cast for three-dimensional rendering. Plane imaging, even if using rays in a volume to find the plane, is not three-dimensional rendering. Smith, et al. do not disclose rays cast as part of volume rendering and do not disclose three-dimensional rendering rays cast through the virtual volume.

Even if Smith, et al. provide for volume rendering, such as at column 26, the approach described does not provide the claim limitations. As noted by the inventor:

> *"Smith describes exactly the approach our [application] tries to avoid because it is too slow and processing intensive. It describes a system that scan converts all the data upfront by computing "any arbitrary three dimensional slice deck through the Ultrasound acquired data with programmable width, length, height and resolution" and "after all slices (defined by dW) of all horizontal points (defined by dU) then initial horizontal point is retrieved and dXdV, dYdV, dZdV parameters are applied to generate the first point of the next scan line. This process continues untill al lines (defined by dV) have been scan converted." (from text in column 26, lines 28-41)*

> *This is brute-force scan conversion. The only difference from other patents the Examiner has brought up before is that in Smith's patent, it is possible to scan convert from any arbitrary 3D orientation. It is a brute-force ray casting approach where all the "rays" forming the arbitrary slice deck are scan converted in their entirety.*
>
> *In our application we avoid scan converting all samples of all rays: we stop as soon as an opaque surface is found so that there is no need to scan convert samples that are occluded and not visible from a particular viewing direction."*

In column 26, Smith, et al. do not disclose the processor is configured to avoid scan-conversion from the polar coordinates of volume data that does not contribute to a final volume rendered image by scan converting only visible voxels based on the rendering rays and corresponding Cartesian coordinates, the identifying corresponding to identifying Cartesian coordinates associated with visible voxels of the final volume rendered image.

Smith, et al. also do not use rays cast in a virtual volume free of voxel data. *Smith, et al. gather data for the volume by scanning (col. 3, lines 55-65). 3D echo data is formed (col. 4, lines 14-16). The echo data represents beams in three dimensions (col. 4, lines 36-39). The use of 3D linear rays by Smith, et al. at column 20, line 65 is for scan conversion of the data (col. 19, lines 41-48). The scan conversion interpolates the echo data (col. 20, lines 12-17). Smith, et al. scan convert the data using 3D linear rays in the data , not rays in a virtual volume free of voxel data. Rays in a virtual volume are not disclosed.*

Independent claim 14 is allowable for similar reasons as claim 1.

Independent claim 28 recites a virtual Cartesian volume being free of voxel values. As noted above for claim 1, Smith, et al. do not use a virtual volume free of voxel values. Smith, et al. provide a scan volume of voxel values. *The use of rays by Smith, et al. is in the actual data. A virtual volume free of voxels is not provided.*

Claim 28 recites volume rendering. Smith, et al. generate slice or sector images, not volume rendering. The use of viewport is not for a rendering, but instead for slices (col. 26, lines 39-41). *As noted above, dx, dy, and dz are used to define a slice in three-dimensions, not as a teaching of volume rendering. Smith, et al. begin the discussion by identifying orthogonal slices to be viewed by the user (see Figures 1a-1i). The user interface 210 provides input to the scan converter to define a slice to be viewed (col. 11, lines 57-65 and col. 19, lines 41-48). "As the raster scan lines move down the frame, the X, Y, Z point traverses the dxdv, dydv, dzdv direction to* <u>*generate any arbitrary slice*</u> *in 3D space" (col. 21, lines 3-6, emphasis by Applicant). The use of scan lines for display is for the creation of each of several arbitrary slices to find an aiming line. The use of multiple slices at column 26 is for cubic interpolation to the view slice (see Figures 17 and 18). The context and described usage of the scan converter discussed at column 26 is for generating two-dimensional images of different arbitrary slices in the volume.*

The "additional state machine mode" at col. 26, lines 49-54 is confusing at best. Given that the scan converter is described as a device for implementing the slice viewing taught in the beginning of the description for the desired purpose of finding an aiming line using slices, the reference to "rendering" is confusing. There is no description of three-dimensional representations or other three-dimensional rendering discussion. Why discuss volume rendering when the goal and previous discussions are for imaging slices in a volume to allow positioning of a line? This may be a cut-n-paste error. If this incomplete and unclear statement is directed to volume rendering, the use of a virtual volume is not provided. To the extent volume rendering is provided in this mode, there is not provided avoidance of scan conversion of voxels that do not contribute to the final volume rendered image. The mode is associated with stepping through the slices. If not to limit to slices for interpolation to the given plane rather than volume rendering, then there is not avoidance of scan conversion. The column 20 discussion is for a different mode of operation.

Claim 28 recites selectively determining, during the volume rendering, the spatial coordinates of the virtual Cartesian volume that are visible in order to avoid scan-conversion computations of acquired ultrasound data associated with polar coordinates for non-visible locations, only the acquired ultrasound data associated with the spatial coordinates of the virtual Cartesian volume that are visible contributing to a given view of the volume being scan converted, the determination of visibility being a function of rays passing through the virtual Cartesian volume from an observer location. Smith, et al. do not consider visibility in the rendering context or as a function of rays passing through the volume from an observer location. *Smith, et al., even if teaching rendering, do not provide for avoidance of scan conversion. The slices are scan converted in the mode at column 26, lines 49-54 without mention of avoiding scan conversion of some voxels. The process merely proceeds through the delta depth for a large number of slices. The only avoidance of scan converting voxels would be with selecting data for a plane. There is no avoidance in general rendering.*

Independent claim 29 is allowable for similar reasons as claim 28.

Dependent claims 2-7, 9, 11-13, 15-20, 22, and 24-27 depend from claims 1 and 14, and are allowable for the same reasons as the corresponding base claim. Further limitations patentably distinguish from the cited references.

Claims 5 and 18 recite volume rendering. Cited col. 4, lines 46-53 note selection of a sub-volume in 3D operation, but do not describe volume rendering. The imaging is slice imaging (see Fig. 1C). Smith, et al. do not disclose rendering of the volume, just steering to a sub-volume for 3D scanning. The 3D echo data is used for imaging arbitrary slices. *As noted above, Smith, et al. desire slice images to aim a line, not volume rendering. To the extent volume rendering is taught, it is an incomplete, confusing, and non-enabled teaching.*

Claims 6 and 19 recite alpha blending. Averaging for interpolation (col. 20, lines 1-20) is not alpha blending. Interpolation is a different process than alpha blending. It is not inherent in interpolation to alpha blend. *In this case, cubic interpolation is used to find each point on a plane. This is not alpha blending. Alpha blending is not simple averaging of a group of points distributed equally in 3D space as in cubic interpolation.* Claim 6 recites use of a threshold for blending. Smith, et al. use bilinear or cubic interpolation (2 or 4). This is not a threshold for blending. Alpha blending and blending in volume rendering are known terms. Interpolation for scan conversion is not alpha blending or thresholding. *Identifying 2 or 4 values closest to a point on a line for interpolation is not use of a threshold to avoid scan conversion.*

Claim 11 recites a GPU. Smith, et al. use a dedicated scan converter, not a GPU. *A graphics processing unit is a term of art, much as a scan converter is a term of art. A scan converter is not a graphics processing unit, despite the scan converter processing image data.*

Claim 12 recites a flag, and an integer sum. The Examiner alleges no criticality and mere design choice. As noted in the specification, an integer sum allows indication of spatial relationship relative to other table entries. The claimed invention is about scan conversion, which is a spatial relationship. The flag and integer sum efficiently track visibility for the claimed reverse lookup. Given the teachings of Smith, et al. and Park, a person of ordinary skill in the art would not have used the flag and integer sum as a design choice. *Park does not provide a mechanism for tracking visibility in the look-up table, so would be inefficient.*

Claim 27 recites volume rendering. The cited portion at col. 1, line 59-col. 2, line 18 shows scanning a volume, but not rendering. For example, CW displays a spectrum and M-mode is a one-dimensional display in space with time as the other dimension. Smith, et al. provide for imaging arbitrary slices in the scanned volume, not volume rendering.

CONCLUSION:

Applicants respectfully submit that all of the pending claims are in condition for allowance and seeks early allowance thereof.

PLEASE MAIL CORRESPONDENCE TO:

Siemens Corporation
Customer No. 28524
Attn: Elsa Keller, Legal Administrator
170 Wood Avenue South
Iselin, NJ 08830

Respectfully submitted,
/Jenny G. Ko/

Jenny G. Ko, Reg. No. 44,190
Attorney(s) for Applicant(s)
Telephone: 650-694-5810
Date: November 3, 2010